Deep Neural Net Approaches for Natural Language Processing

Gary Geunbae Lee
Dept of CSE/AIGS
POSTECH
Deep Learning Basics
AI vs. ML vs. DL

Artificial Intelligence
- Reasoning
- Knowledge Representation
- Planning
- Learning
- Perception
- Manipulation

Machine Learning
- Supervised Learning with Teachers
- Semi-supervised Learning with Teacher
- Unsupervised Learning without Teacher
- Reinforcement Learning with Rewards

Deep Learning
- RBM (Restricted Boltzmann Machine)
- DBN (Deep Belief Network)
- CNN (Convolutional Neural Network)
- Deep Reinforcement Learning

Input: Text, Sound, Image, Video

Output: Generated Information / Autonomous Control
Artificial Neural Networks (ANN)

\[ \sum x_1 x_2 w_1 w_2 w_3 w_n \]

Dendrites

Terminal Branches of Axon

Activation Function

Axon
Layered Networks

\[ y = \sum_{i,j} f(w_{ij}x_j) \]

Output:
\[ y = f(w^1x + w^2x + w^3x + \cdots + w^mx_m) \]
\[ = f(\sum_j w^j_i x_j) \]
Deep learning Innovation

• Combining Feature Learning and Classification as Unified Framework (※ Learning what to learn, how to learn)

Feature learning aspect of DNN based Image Classification
Vanilla recurrent neural networks (RNNs)

- RNNs have connections from the outputs of previous time steps to inputs of next time steps.

- For sequential data, a RNN usually computes hidden state $h_t$ from the previous hidden state $h_{t-1}$ and the input $x_t$.
  \[
  h_t = \sigma(W_h h_{t-1} + W_x x_t + b)
  \]
Vanishing gradient problem

\[ h_t = \sigma(W_h h_{t-1} + W_x x_t + b) \]

- Let’s assume \( \sigma \) is the identity function

\[
\frac{\partial J^{(i)}(\theta)}{\partial h^{(j)}} = \frac{\partial J^{(i)}(\theta)}{\partial h^{(i)}} \prod_{j < t \leq i} \frac{\partial h^{(t)}}{\partial h^{(t-1)}}
\]
\[
= \frac{\partial J^{(i)}(\theta)}{\partial h^{(i)}} \prod_{j < t \leq i} W_h = \frac{\partial J^{(i)}(\theta)}{\partial h^{(i)}} W_h^\ell
\]

If all \( \frac{\partial h^t}{\partial h^{t-1}} < 1 \) \( \Rightarrow \frac{\partial J^t}{\partial h^1} \approx 0 \)
Long short-term memory networks (LSTMs)

- LSTMs explicitly keep and update cell memory $c^{(t)}$ by
  - Removing the previous cell content $c^{(t-1)}$ by multiplying it with $f^{(t)}$
  - Adding the new cell content $\tilde{c}^{(t)}$ multiplied by $i^{(t)}$
- LSTMs produce output $h^{(t)} = o^{(t)} \circ \tanh c^{(t)}$

\[
\begin{align*}
f^{(t)} &= \sigma \left( W_f h^{(t-1)} + U_f x^{(t)} + b_f \right) \\
i^{(t)} &= \sigma \left( W_i h^{(t-1)} + U_i x^{(t)} + b_i \right) \\
o^{(t)} &= \sigma \left( W_o h^{(t-1)} + U_o x^{(t)} + b_o \right) \\
\tilde{c}^{(t)} &= \tanh \left( W_c h^{(t-1)} + U_c x^{(t)} + b_c \right) \\
c^{(t)} &= f^{(t)} \circ c^{(t-1)} + i^{(t)} \circ \tilde{c}^{(t)} \\
h^{(t)} &= o^{(t)} \circ \tanh c^{(t)}
\end{align*}
\]
Gated recurrent units (GRUs)

• GRUs keeps and update $h(t)$ by two gates:
  • Update gate $u(t)$ decides
    • How much the old hidden representation $h(t)$ is removed
    • how much the new hidden representation $\tilde{h}(t)$ is added
  • Reset gate $r(t)$ decides how much old representation $h(t)$ is needed to compute new representation $\tilde{h}(t)$

• GRUs also use less number of gates and have smaller parameters than LSTMs

\[
\begin{align*}
    u(t) &= \sigma \left( W_u h(t-1) + U_u x(t) + b_u \right) \\
    r(t) &= \sigma \left( W_r h(t-1) + U_r x(t) + b_r \right) \\
    \tilde{h}(t) &= \tanh \left( W_h (r(t) \circ h(t-1)) + U_h x(t) + b_h \right) \\
    h(t) &= (1 - u(t)) \circ h(t-1) + u(t) \circ \tilde{h}(t)
\end{align*}
\]
Bidirectional Multi-Layer RNNs

the
movie
was
terribly
exciting
!
Parallel computing for Deep Learning

- History of parallel/distributed systems for Deep Learning computing

Google taps 16k computers to look for cats—for Science!

- Univ. of Toronto uses 2 GPUs for 1.2M training images for 1000 classes image classification (※ ImageNet Large Scale Visual Recognition Challenge)

- Stanford uses 12 GPUs for large-scale video classification with convolutional neural networks (※ 10M Youtube video)

- Google uses 16K CPU cores for training 22-layers deep neural network (※ GoogLeNet, 2014)

- Baidu’s Artificial Intelligence Supercalculator beats Google at Image Recognition
Word Vector

- Represent words as vectors

\[ \text{expect} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{pmatrix} \]
Word Vector

• Distributional semantics: A word’s meaning is given by the words that frequently appear close-by

  “You shall know a word by the company it keeps”

• Word2vec objective function (skip-grams)

\[
J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P(w_{t+j} | w_t; \theta)
\]
• A word’s **contextual embedding** must consider its context
ELMo: Embeddings from Language Model

- Multi-layer bidirectional LSTM language model

\[ \gamma_{task} : \text{scale (hyper-parameter)} \]
\[ s^\text{task}_j : \text{weight (learned)} \]
ELMo for MRC

• ELMo as a word embedding
Transformer

- Parallel self-attention
  - Looks at self, and determines where to focus

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V
\]

Q,K,V – vectors for every word, output attention summed up; head means different Q,K,V vector with different weights

Vaswani, Ashish, et al. "Attention is all you need." *NIPS 2017*
• Training 1. Masked words prediction
  • 15% of words are [MASK]ed

*GELU: Gaussian error linear unit
BERT: Bidirectional Encoder Representations from Transformers

- Training 2. Next sentence prediction
  - To understand texts more than a sentence

**Input** = [CLS] the man went to [MASK] store [SEP]

he bought a gallon [MASK] milk [SEP]

**Label** = IsNext

**Input** = [CLS] the man [MASK] to the store [SEP]

penguin [MASK] are flight #less birds [SEP]

**Label** = NotNext
BERT: Bidirectional Encoder Representations from Transformers

- BERT as universal pre-trained model for NLP
  - BERT requires minimal additional layers and fine-tuning

GPT (Generative Pre-Training)

In pre-training, optimize $L_1(u)$

$u$: Unlabeled dataset

$\Theta$: Model parameters

In fine-tuning, optimize $L_3(c)$

$c$: Labeled dataset

$\lambda$: Hyper-parameter weight

*$GPT-2/3$: zero/few-shot learning
Sequence labeling

- Sequence labeling is the task of assigning a categorical label to each member of an observed sequence.

Examples of sequence labeling:

- **Part-of-speech tagging** labels each word with a grammatical category.
  - e.g. The | trees | are | ... → DT (determiner) | NNS (plural noun) | VBP (plural verb)

- **Named entity recognition** locates and classifies named entity in text. It can be tackled by labeling each word with a named entity category.
  - e.g. Barack | Obama | said | ... → B-PERSON | I-PERSON | O | ...
Bi-directional LSTM-CNNs-CRF

- **Bi-directional LSTMs** encode word embeddings and character representations.

- **Conditional random fields** compute the distribution of output sequence:
  - Viterbi algorithm is applied during training and decoding.
  - The objective is the negative log-likelihood of the output sequence distribution.

\[
p(y|z; W, b) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, z)}{\sum_{y' \in \mathcal{Y}(z)} \prod_{i=1}^{n} \psi_i(y'_{i-1}, y'_i, z)}
\]

- **z**: input sequence.
- **y**: output sequence.
- **\( \mathcal{Y}(z) \)**: a set of all possible output sequences when given the input sequence **z**.

[Ma 2016]
Seq2Seq NMT via fixed-length representations

- Encoder RNN compresses input sequence into a fixed-length representation
- Decoder RNN produces output sequence from the representation
  - Each produced output token is fed into the next RNN’s input

[Sutskever 2014]
S2S NMT with attention mechanism

• It’s hard to encode all the information of an input sequence into a fixed-length representation

• We can focus important parts of input sequences for each decoding step by attention mechanism

[Bahdanau 2015]
“Multilingual Neural Machine Translation with **Knowledge Distillation**”

(Microsoft Research, ICLR 2019)

- **Knowledge Distillation**
  - Use the knowledge of big & well-trained networks when training small model
“Multilingual Neural Machine Translation with Knowledge Distillation”

• Applying Knowledge Distillation by adjusting Loss Function

• Loss Function

\[
L_{\text{ALL}}(D; \theta, \theta_T) = (1 - \lambda)L_{\text{NLL}}(D; \theta) + \lambda L_{\text{KD}}(D; \theta, \theta_T),
\]

* Selective Distillation
*Better than interpolation
: only when ”Individual accuracy > Multilingual accuracy” apply Lkd loss, else apply LNLL loss

\[
L_{\text{NLL}}(D; \theta) = - \sum_{(x,y) \in D} \log P(y|x; \theta),
\]

\[
\log P(y|x; \theta) = \sum_{t=1}^{T_y} \sum_{k=1}^{|\mathcal{V}|} \mathbb{1}\{y_t = k\} \log P(y_t = k|y_{<t}, x; \theta),
\]

\[
L_{\text{KD}}(D; \theta, \theta_T) = - \sum_{(x,y) \in D} \sum_{t=1}^{T_y} \sum_{k=1}^{|\mathcal{V}|} Q\{y_t = k|y_{<t}, x; \theta_T\} \log P(y_t = k|y_{<t}, x; \theta).
\]
MNMT (Multilingual Neural Machine Translation)

- Based on Transformer Encoder-Decoder Model

Ko-En topk probability
Ja-En topk probability
Vi-En topk probability
Zh-En topk probability

Ko → En
Ja → En
Vi → En
Zh → En

Ko → En
Vi → En
Dependency parsing

- Dependency parsing is the task of extracting **dependencies** between **head** and **dependent** words from a sentence.

- A dependency is the arrow from a head to a dependent with a grammatical type called **relation** (e.g. nsubj).

- Dependencies show which words depend on (modify or are arguments of) which other words.
Neural transition-based dependency parsing

- Extract features from Stack and Buffer
  - lc/rc: leftmost/rightmost children
- Classify an action by neural networks
  - The objective is the negative log-likelihood of the action distribution

Feature extraction

Feedforward neural network-based action classifier

[Chen 2014]
Semantic parsing is a task of mapping natural language to programs.

We aim to develop semantic parsers without direct supervision on programs.

Program:
(map (argmax (filter all-rows (λ (x) (= (string:country x) "greece"))) index) number:year)

Natural language:
“Greece held its last Summer Olympics in which year?”

Denotation:
2004

Context:

<table>
<thead>
<tr>
<th>Year</th>
<th>City</th>
<th>Country</th>
<th>Nations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1896</td>
<td>Athens</td>
<td>Greece</td>
<td>14</td>
</tr>
<tr>
<td>1900</td>
<td>Paris</td>
<td>France</td>
<td>24</td>
</tr>
<tr>
<td>1904</td>
<td>St. Louis</td>
<td>USA</td>
<td>12</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2004</td>
<td>Athens</td>
<td>Greece</td>
<td>201</td>
</tr>
<tr>
<td>2008</td>
<td>Beijing</td>
<td>China</td>
<td>204</td>
</tr>
<tr>
<td>2012</td>
<td>London</td>
<td>UK</td>
<td>204</td>
</tr>
</tbody>
</table>
• Bottom-up neural semantic parsing
  • Neural networks decodes logic actions
  • Grammar of logical form is restricted (function definition)

\[
\begin{align*}
(Hop \ r \ p) & \Rightarrow \{e_2 | e_1 \in r, (e_1, p, e_2) \in K\} \\
(ArgMax \ r \ p) & \Rightarrow \{e_1 | e_1 \in r, \exists e_2 \in E : (e_1, p, e_2) \in K, \forall e : (e_1, p, e) \in K, e_2 \geq e\} \\
(ArgMin \ r \ p) & \Rightarrow \{e_1 | e_1 \in r, \exists e_2 \in E : (e_1, p, e_2) \in K, \forall e : (e_1, p, e) \in K, e_2 \leq e\} \\
(Filter \ r_1 \ r_2 \ p) & \Rightarrow \{e_1 | e_1 \in r_1, \exists e_2 \in r_2 : (e_1, p, e_2) \in K\}
\end{align*}
\]

Table 1: Interpreter functions. \(r\) represents a variable, \(p\) a property in Freebase. \(\geq\) and \(\leq\) are defined on numbers and dates.

Figure 2: Semantic Parsing with NSM. The key embeddings of the key-variable memory are the output of the sequence model at certain encoding or decoding steps. For illustration purposes, we also show the values of the variables in parentheses, but the sequence model never sees these values, and only references them with the name of the variable (“\(R_1\)”). A special token “GO” indicates the start of decoding, and “Return” indicates the end of decoding.

Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision, Liang et al., ACL 2017

Memory Augmented Policy Optimization for Program Synthesis and Semantic Parsing, Liang et al., NIPS 2018
• Top-down neural semantic parsing
  • Neural net generate derivation actions
  • Type system reduce search space (entity matching)

Neural Semantic Parsing with Type Constraints for Semi-Structured Tables, Krishnamurthy et al., EMNLP 2017

Iterative Search for Weakly Supervised Semantic Parsing, Dasigi et al., NAACL 2019
code generation by semantic parsing

\[
\text{root} \Rightarrow \text{Expr} \\
\text{Expr} \Rightarrow \text{expr}[\text{value}] \\
\text{expr} \Rightarrow \text{Call} \\
\text{Call} \Rightarrow \text{expr}[\text{func} \text{ expr*}[\text{args}] \text{ keyword*}[\text{keywords}]] \\
\text{expr} \Rightarrow \text{Name} \\
\text{Name} \Rightarrow \text{str} \\
\text{GenToken}[\text{sorted}] \\
\text{GenToken}[<\text{/n}>] \\
\text{expr*} \Rightarrow \text{expr} \\
\text{Name} \Rightarrow \text{str} \\
\text{GenToken}[\text{my_list}] \\
\text{GenToken}[<\text{/n}>] \\
\text{keyword*} \Rightarrow \text{keyword} \\
\text{keyword} \Rightarrow \text{str} \\
\text{Action Flow} \\
\text{Parent Feeding} \\
\text{Apply Rule} \\
\text{Generate Token} \\
\text{GenToken with Copy} \\
\]

**ASDL Grammar**

\[
\text{stmt} \Rightarrow \text{Expr}(\text{expr value}) \\
\text{expr} \Rightarrow \text{Call}(\text{expr func, expr* [args], keyword* [keywords]}) \\
\text{Attribute}(\text{expr value, identifier attr}) \\
\text{Name}(\text{identifier id}) \\
\text{Str}(\text{string s})
\]

**Abstract Syntax Description Language (ASDL) for Python**

\[
\text{Code}: \text{sorted(my_list, reverse=True)}
\]

\[
\text{stmt} = \text{Select}(\text{agg_op? agg, idx column_idx, cond_expr* conditions}) \\
\text{cond_expr} = \text{Condition}(\text{cmp_op op, idx column_idx, string value}) \\
\text{agg_op} = \text{Max} | \text{Min} | \text{Count} | \text{Sum} | \text{Avg} \\
\text{cmp_op} = \text{Equal} | \text{GreaterThan} | \text{LessThan} | \text{Other}
\]

[\text{Yin 2017, Yin 2018, Rabinovich 2017}]
Sentiment Analysis

- interpretation and classification of emotions (positive, negative and neutral) within text data using text analysis techniques
Sentiment Analysis

- XLNet based classification

*XLNet = GPT (AR)+BERT(AE): permutation AR (Transformer-XL)
ConvNet for NLP

RNN for NLP - softmax is often only calculated at the last step

CNN for NLP
ConvNet for NLP

- CNN architecture for sentence classification

Text classification by CNN+LSTM

- CNN: advantages in selecting good features
- LSTM networks: good abilities of learning sequential data.
Style-based fake news detection

Wording, writing style

1) Use transfer learning
: improve the performance of the model by training other related tasks before training target task
: Use ELMo for transfer learning

2) Use CNN+BiLSTM+LSTM
for malicious comments detection

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>no transfer learning</td>
<td>0.9503</td>
<td>0.8949</td>
<td>0.9495</td>
<td>0.9214</td>
</tr>
<tr>
<td>ELMo</td>
<td>0.9669</td>
<td>0.9269</td>
<td>0.9686</td>
<td>0.9473</td>
</tr>
</tbody>
</table>

=> Better performance when using ELMO as transfer learning

https://d2.naver.com/helloworld/7753273
Cross-Lingual Adversarial Method

ROAD (Reference Oriented Adversarial Fake News Detector)
• Unanswerable question (negative example)
  • Relevant to the topic
  • Existence of plausible answers

**Article:** Endangered Species Act
**Paragraph:** “…Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society—the species were relatively rare—and little opposition was raised.”

**Question 1:** “Which laws faced significant opposition?”
**Plausible Answer:** later laws

**Question 2:** “What was the name of the 1937 treaty?”
**Plausible Answer:** Bald Eagle Protection Act

Figure 1: Two unanswerable questions written by crowdworkers, along with plausible (but incorrect) answers. Relevant keywords are shown in blue.
SQuAD2.0

- BERT – no answer prediction

answerable case

unanswerable case
Reasoning MRC Dataset

✅ Quoref (Questions Requiring Coreferential Reasoning)
- Containing 24K questions over 4.7K paragraphs from Wikipedia.
- Resolve hard coreferences before selecting the appropriate span(s) in the paragraphs for answering questions.

✅ DROP (Discrete Reasoning Over Paragraphs)
- Wikipedia-based 96K QA pairs over 6,700 paragraphs.
- Most of the questions are mainly based on numerical arithmetic.

<table>
<thead>
<tr>
<th>Reasoning</th>
<th>Passage (some parts shortened)</th>
<th>Question</th>
<th>Answer</th>
<th>BiDAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtraction (28.8%)</td>
<td>That year, his Untitled (1981), a painting of a haloed, black-headed man with a bright red skeletal body, depicted amid the artists signature scrawls, was sold by Robert Lehrman for $16.3 million, well above its $12 million high estimate.</td>
<td>How many more dollars was the Untitled (1981) painting sold for than the $12 million dollar estimation?</td>
<td>4300000</td>
<td>$16.3 million</td>
</tr>
<tr>
<td>Comparison (18.2%)</td>
<td>In 1517, the seventeen-year-old King sailed to Castle. There, his Flemish court ..., In May 1518, Charles traveled to Barcelona in Aragon.</td>
<td>Where did Charles travel to first, Castle or Barcelona?</td>
<td>Castle Aragon</td>
<td></td>
</tr>
<tr>
<td>Selection (19.4%)</td>
<td>In 1970, to commemorate the 100th anniversary of the founding of Baldwin City, Baker University professor and playwright Don Mueller and Phyllis E. Brown, Business Manager, produced a musical play entitled The Ballad Of Black Jack to tell the story of the events that led up to the battle.</td>
<td>Who was the University professor that helped produce The Ballad Of Black Jack, Ivan Boyd or Don Mueller?</td>
<td>Don Mueller Baker</td>
<td></td>
</tr>
<tr>
<td>Addition (11.7%)</td>
<td>Before the UNPROFOR fully deployed, the HV clashed with an armed force of the RSF in the village of Nos Kalić, located in a pink zone near Šibenik, and captured the village at 4:45 p.m. on 2 March 1992. The JNA formed a battalion to counterattack the next day.</td>
<td>What date did the JNA form a battlegroup to counterattack after the village of Nos Kalić was captured?</td>
<td>3 March 1992</td>
<td>2 March 1992</td>
</tr>
<tr>
<td>Count (16.5%) and Sort (11.7%)</td>
<td>Denver would retake the lead with kicker Matt Prater nailing a 43-yard field goal, yet Carolina answered as kicker John Kasay ties the game with a 39-yard field goal, ...</td>
<td>Which kicker kicked the most field goals?</td>
<td>John Kasay Matt Prater</td>
<td></td>
</tr>
<tr>
<td>Other Arithmetic (3.2%)</td>
<td>Although the movement initially gathered some 60,000 adherents, the subsequent establishment of the Bulgarian Exarchate reduced their number by some 75 %.</td>
<td>How many adherents were left after the establishment of the Bulgarian Exarchate?</td>
<td>15000</td>
<td>60,000</td>
</tr>
<tr>
<td>Set of spans (6.9%)</td>
<td>According to some sources 363 civilians were killed in Kavadarci, 230 in Negotin and 40 in Vatasha.</td>
<td>What does the 3 villages that people were killed in?</td>
<td>Kavadarci, Negotin, Vatasha</td>
<td></td>
</tr>
<tr>
<td>Other (6.8%)</td>
<td>This Annual Financial Report is our principal financial statement of accountability. The AFR gives a comprehensive view of the Department’s financial activities ...</td>
<td>What does AFR stand for?</td>
<td>Annual Financial Report</td>
<td>one of the Big Four audit firms</td>
</tr>
</tbody>
</table>
Multiple span type prediction

- Question embedding used to direct the reasoning in graph reasoning.

Answer type prediction

- The answers to the questions are categorized into 5 types.

- Single span type prediction:
  - Start and end probability for each token is computed.

- Multiple span type prediction:
  - Each token is classified by I/O tagging method, and multiple tokens classified as I are determined as answers.

- Classification for count type prediction:
  - 10 class classification problem (0~9), which covers about 97% counting problem in the DROP dataset.

- Classification for add/sub type prediction:
  - Only addition and subtraction operations are involved, each number is classified into one of (-1, 0, +1)

Question Directed Graph Attention Network

- Relation information between numbers and entities in the passages is updated through graph reasoning (self-attention mechanism).

RoBERTa Encoder

- Question and passage tokens are encoded by large RoBERTa encoder (max sequence: 512).
Graph Reasoning Module

Node
✓ Entities (noun & pronoun) & numbers in sentences
  - Each number type is categorized into 7 number types.
    (CARDINAL, PERCENT, MONEY, TIME, DATE, ORDINAL, QUANTITY)

Edge = attention strength
✓ The numbers of the same type are connected with each other by the type-specific edge.
✓ The entities and the numbers are connected when they co-occur in a sentence.

Graph Attention Network

1. Numerical Reasoning Module
   → The passage and question representation from RoBERTa's output are linearly transformed and input into the graph reasoning module.

\[
M^Q = W^M Q,
M^P = W^P P,
\]
\[
c = W^\text{MEAN}(Q),
U = QDGAT(G; M^P, M^Q, c)
\]

2. Question Directed Node Embedding Update
   → Query, key, value are calculated for graph attention

\[
v_i = M^P [I^P(v_i)]
\]
\[
m^t = W^d_{xv}(W_{fj} c),
\]
\[
x_{i}^t = W_{xv}[v^t : v^0] \odot W_{xv} m^t,
\]
\[
x_{k}^t = W_{vk}[v^t : v^0] \odot W_{vk} m^t,
\]
\[
x_{v}^t = W_{vv}[v^t : v^0] \odot W_{vv} m^t
\]

3. Directed Graph Attention
   → After applying attention to the representations of neighboring nodes, the representation of the corresponding node is updated.

\[
\alpha_{i,j} = f(\sum_{r \in R_{i,j}} \exp(a_{i,j}^r))
\]
\[
\alpha_{i,j}^r = \sum_{j' \in V_{i,j}} \exp(a_{i,j}^{r,j'})
\]
\[
\tilde{x}_i^t = \sum_{j \in N_i} \alpha_{i,j} x_{i,j},
\]
\[
v_{i}^{t+1} = W^u[v_i^t; \tilde{x}_i^t],
\]
\[
v_{v}^{t+1} = QDGAT-single(G, v^t, c)
\]

4. Module Output
   → Representations of entities and numbers are updated.

\[
U_i = \begin{cases} M_i + v_{i}^{T(i)} & \text{if } i\text{-th token } \in V \\ M_i, & \text{otherwise} \end{cases}
\]

*Only number, entity updated

Q: Question
P: passage

Question directed graph attention network (step 2,3 repeated)
: Self-attention in graph
- large-scale dataset which contains questions that depend on a conversation history
- Answers can be **free-form text**, not text spans from the passages (first select a text span as the rationale and then edit it to obtain a free-form answer)
- Seven domains passage

### 2. Dataset

<table>
<thead>
<tr>
<th>Domain</th>
<th>#Passages</th>
<th>#Q/A pairs</th>
<th>Passage length</th>
<th>#Turns per passage</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-domain</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Children’s Sto.</td>
<td>750</td>
<td>10.5k</td>
<td>211</td>
<td>14.0</td>
</tr>
<tr>
<td>Literature</td>
<td>1,815</td>
<td>25.5k</td>
<td>284</td>
<td>15.6</td>
</tr>
<tr>
<td>Mid/High Sch.</td>
<td>1,911</td>
<td>28.6k</td>
<td>306</td>
<td>15.0</td>
</tr>
<tr>
<td>News</td>
<td>1,902</td>
<td>28.7k</td>
<td>268</td>
<td>15.1</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>1,821</td>
<td>28.0k</td>
<td>245</td>
<td>15.4</td>
</tr>
<tr>
<td><em>only test data</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reddit</td>
<td>100</td>
<td>1.7k</td>
<td>361</td>
<td>16.6</td>
</tr>
<tr>
<td>Science</td>
<td>100</td>
<td>1.5k</td>
<td>251</td>
<td>15.3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>8,399</td>
<td>127k</td>
<td>271</td>
<td>15.2</td>
</tr>
</tbody>
</table>

Table 2: Distribution of domains in CoQA.

<table>
<thead>
<tr>
<th></th>
<th>SQuAD</th>
<th>CoQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passage Length</td>
<td>117</td>
<td>271</td>
</tr>
<tr>
<td>Question Length</td>
<td>10.1</td>
<td>5.5</td>
</tr>
<tr>
<td>Answer Length</td>
<td>5.2</td>
<td>2.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>SQuAD</th>
<th>CoQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answerable</td>
<td>66.7%</td>
<td>98.7%</td>
</tr>
<tr>
<td>Unanswerable</td>
<td>33.3%</td>
<td>1.3%</td>
</tr>
<tr>
<td>Span found</td>
<td>100.0%</td>
<td>66.8%</td>
</tr>
<tr>
<td>No span found</td>
<td>0.0%</td>
<td>33.2%</td>
</tr>
<tr>
<td>Named Entity</td>
<td>35.9%</td>
<td>28.7%</td>
</tr>
<tr>
<td>Noun Phrase</td>
<td>25.0%</td>
<td>19.6%</td>
</tr>
<tr>
<td>Yes</td>
<td>0.0%</td>
<td>11.1%</td>
</tr>
<tr>
<td>No</td>
<td>0.1%</td>
<td>8.7%</td>
</tr>
<tr>
<td>Number</td>
<td>16.5%</td>
<td>9.8%</td>
</tr>
<tr>
<td>Date/Time</td>
<td>7.1%</td>
<td>3.9%</td>
</tr>
<tr>
<td>Other</td>
<td>15.5%</td>
<td>18.1%</td>
</tr>
</tbody>
</table>

Table 3: Average number of words in passage, question and answer in SQuAD and CoQA.

Figure 1: A conversation from the CoQA dataset. Each turn contains a question ($Q_i$), an answer ($A_j$) and a rationale ($R_i$) that supports the answer.
1. Characteristics

- Coreference to previous questions and answers
- **Open-ended questions** (Yes, No, unknown answers)
- Dialog action (teacher -> student)
  1. continuation (follow up \(\leftrightarrow\), maybe follow up \(\leftrightarrow\), don’t follow up \(\not\leftrightarrow\))
  2. affirmation (yes, no, neither)
  3. answerable (answerable, no answer)
- Answer = (answer span, dialog action)

2. Dataset (**Metrics: F1, HEQ-Q, HEQ-D**)

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Dev.</th>
<th>Test</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>questions</td>
<td>83,568</td>
<td>7,354</td>
<td>7,353</td>
<td>98,407</td>
</tr>
<tr>
<td>dialogs</td>
<td>11,567</td>
<td>1,000</td>
<td>1,002</td>
<td>13,594</td>
</tr>
<tr>
<td>unique sections</td>
<td>6,843</td>
<td>1,000</td>
<td>1,002</td>
<td>8,854</td>
</tr>
<tr>
<td>tokens / section</td>
<td>396.8</td>
<td>440.0</td>
<td>445.8</td>
<td>401.0</td>
</tr>
<tr>
<td>tokens / question</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
<td>6.5</td>
</tr>
<tr>
<td>tokens / answer</td>
<td>15.1</td>
<td>12.3</td>
<td>12.3</td>
<td>14.6</td>
</tr>
<tr>
<td>questions / dialog</td>
<td>7.2</td>
<td>7.4</td>
<td>7.3</td>
<td>7.2</td>
</tr>
<tr>
<td>% yes/no</td>
<td>26.4</td>
<td>22.1</td>
<td>23.4</td>
<td>25.8</td>
</tr>
<tr>
<td>% unanswerable</td>
<td>20.2</td>
<td>20.2</td>
<td>20.1</td>
<td>20.2</td>
</tr>
</tbody>
</table>

The human equivalence score (HEQ): judge whether a system’s output is as good as that of an average human
- **HEQ-Q**: the percentage of questions for which this is true
- **HEQ-D**: the percentage of dialogs for which this is true for every question in the dialog

![Example dialog](image)
Focus on handling conversation history to understand and answer the current question

- History attention mechanism (compute the attention weight according to the utility of the history turn in each variation)

- Positional history answer embedding (PosHAE) –
- multi-task learning (MTL)

- MTL (answer span prediction & dialog act prediction)

*history attention mechanism

**Dataset:** QuAC

**Metrics:** F1, HEQ-Q, HEQ-D

<table>
<thead>
<tr>
<th>Models</th>
<th>F1</th>
<th>HEQ-Q</th>
<th>HEQ-D</th>
<th>Yes/No</th>
<th>Follow up</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiDAF++</td>
<td>51.8</td>
<td>50.2</td>
<td>45.3</td>
<td>43.3</td>
<td>2.0/2.2</td>
</tr>
<tr>
<td>BiDAF++ w/ 2-C</td>
<td>60.6</td>
<td>60.1</td>
<td>55.7</td>
<td>54.8</td>
<td>5.3/4.0</td>
</tr>
<tr>
<td>BERT + HAE</td>
<td>63.9</td>
<td>62.4</td>
<td>59.7</td>
<td>57.8</td>
<td>5.9/5.1</td>
</tr>
<tr>
<td>FlowQA</td>
<td>64.6</td>
<td>64.1</td>
<td>-</td>
<td>59.6</td>
<td>-5.8/5.8</td>
</tr>
<tr>
<td>BERT + PosHAE</td>
<td>64.7</td>
<td>-60.7</td>
<td>-6.0</td>
<td>-</td>
<td>N/A</td>
</tr>
<tr>
<td>HAM</td>
<td>65.5</td>
<td>64.4</td>
<td>62.1</td>
<td>60.2</td>
<td>7.3/6.1</td>
</tr>
<tr>
<td>HAM (BERT-Large)</td>
<td>66.7</td>
<td>/65.4</td>
<td>63.3</td>
<td>61.8</td>
<td>9.5/6.7</td>
</tr>
<tr>
<td>Each cell displays val/test scores.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Variation: only one turn of conversation history (q, a pair sequence)

q: question, p: passage, H: history

*Sliding window for long passage

*each line in history
### CoQA

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>In-domain</th>
<th>Out-of-domain</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RL-based + AT + KD (ensemble)</td>
<td>91.4</td>
<td>89.2</td>
<td>90.7</td>
</tr>
<tr>
<td>2</td>
<td>TR-MT (ensemble)</td>
<td>91.5</td>
<td>88.8</td>
<td>90.7</td>
</tr>
<tr>
<td>3</td>
<td>RoBERTa + AT + KD (single model)</td>
<td>90.9</td>
<td>89.2</td>
<td>90.4</td>
</tr>
<tr>
<td>4</td>
<td>Google SQuAD 2.0 + MMFT (ensemble)</td>
<td>89.9</td>
<td>88.0</td>
<td>89.4</td>
</tr>
<tr>
<td>5</td>
<td>AT-MT (single model)</td>
<td>90.4</td>
<td>86.8</td>
<td>89.3</td>
</tr>
<tr>
<td>6</td>
<td>XLNet + Augmentation (single model)</td>
<td>89.9</td>
<td>86.9</td>
<td>89.0</td>
</tr>
<tr>
<td>7</td>
<td>Google SQuAD 2.0 + MMFT (single model)</td>
<td>88.5</td>
<td>86.0</td>
<td>87.8</td>
</tr>
<tr>
<td>8</td>
<td>ConBiERT (ensemble)</td>
<td>88.7</td>
<td>85.4</td>
<td>87.5</td>
</tr>
<tr>
<td>9</td>
<td>BERT + MMFT + ADA (ensemble)</td>
<td>87.5</td>
<td>85.3</td>
<td>86.8</td>
</tr>
<tr>
<td>10</td>
<td>BERT + MMFT + ADA (single model)</td>
<td>86.4</td>
<td>81.9</td>
<td>85.0</td>
</tr>
<tr>
<td>11</td>
<td>XLNet + MMFT + ADA (single model)</td>
<td>85.7</td>
<td>81.7</td>
<td>84.6</td>
</tr>
</tbody>
</table>

### QuAC

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>In-domain</th>
<th>Out-of-domain</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EL-QA (single model)</td>
<td>74.6</td>
<td>71.6</td>
<td>74.1</td>
</tr>
<tr>
<td>2</td>
<td>HistoryQA</td>
<td>74.2</td>
<td>71.5</td>
<td>73.9</td>
</tr>
<tr>
<td>3</td>
<td>TR-MT (ensemble)</td>
<td>74.4</td>
<td>71.3</td>
<td>73.6</td>
</tr>
<tr>
<td>4</td>
<td>RoBERTa + DA (ensemble)</td>
<td>74.0</td>
<td>70.7</td>
<td>73.1</td>
</tr>
<tr>
<td>5</td>
<td>History-Attentive-TransBERT (single model)</td>
<td>72.9</td>
<td>69.7</td>
<td>71.6</td>
</tr>
<tr>
<td>6</td>
<td>RoBERTa + DA (single model)</td>
<td>73.5</td>
<td>69.8</td>
<td>71.2</td>
</tr>
<tr>
<td>7</td>
<td>BERTMT (ensemble)</td>
<td>72.3</td>
<td>69.4</td>
<td>70.9</td>
</tr>
<tr>
<td>8</td>
<td>XLNet + Augmentation (single model)</td>
<td>71.2</td>
<td>67.5</td>
<td>70.4</td>
</tr>
<tr>
<td>9</td>
<td>TransBERT (single model)</td>
<td>71.4</td>
<td>68.1</td>
<td>71.2</td>
</tr>
<tr>
<td>10</td>
<td>BERTMT (single model)</td>
<td>69.4</td>
<td>66.0</td>
<td>68.0</td>
</tr>
<tr>
<td>11</td>
<td>Contest-Aware-BERT (single model)</td>
<td>69.6</td>
<td>65.7</td>
<td>67.9</td>
</tr>
<tr>
<td>12</td>
<td>BertInfoFlow (single model)</td>
<td>69.3</td>
<td>65.2</td>
<td>66.5</td>
</tr>
<tr>
<td>13</td>
<td>ConBiERT (ensemble)</td>
<td>68.0</td>
<td>63.5</td>
<td>65.8</td>
</tr>
<tr>
<td>14</td>
<td>zhzbBERT (single model)</td>
<td>67.0</td>
<td>62.5</td>
<td>66.0</td>
</tr>
<tr>
<td>15</td>
<td>HAM (single model)</td>
<td>65.4</td>
<td>61.8</td>
<td>63.6</td>
</tr>
</tbody>
</table>

### SQuAD2.0

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Human Performance</td>
<td>86.831</td>
<td>89.452</td>
</tr>
<tr>
<td>2</td>
<td>SA-Net on Albert (ensemble)</td>
<td>90.724</td>
<td>93.011</td>
</tr>
<tr>
<td>3</td>
<td>Retro-Reader (ensemble)</td>
<td>90.578</td>
<td>92.978</td>
</tr>
<tr>
<td>4</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>90.442</td>
<td>92.877</td>
</tr>
<tr>
<td>5</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>90.442</td>
<td>92.839</td>
</tr>
<tr>
<td>6</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>90.420</td>
<td>92.799</td>
</tr>
<tr>
<td>7</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>90.454</td>
<td>92.748</td>
</tr>
<tr>
<td>8</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>90.386</td>
<td>92.777</td>
</tr>
<tr>
<td>9</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>90.115</td>
<td>92.580</td>
</tr>
<tr>
<td>10</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>89.002</td>
<td>92.497</td>
</tr>
<tr>
<td>11</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>89.002</td>
<td>92.425</td>
</tr>
<tr>
<td>12</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>89.731</td>
<td>92.215</td>
</tr>
<tr>
<td>13</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>89.743</td>
<td>92.180</td>
</tr>
<tr>
<td>14</td>
<td>ELECTRA+ALBERT+EntitySpaFocus (ensemble)</td>
<td>89.551</td>
<td>92.366</td>
</tr>
</tbody>
</table>
The Stanford Natural Language Inference (SNLI) corpus - https://nlp.stanford.edu/projects/snli/  

- **Size:** 570k sentence pairs  
- manually labeled for balanced classification with the labels *entailment*, *contradiction*, and *neutral*, supporting the task of natural language inference (NLI), also known as recognizing textual entailment (RTE)  

- **SoTA model:** CA-MTL  
- **Metric:** Accuracy
The Multi-Genre Natural Language Inference (MultiNLI) corpus - https://cims.nyu.edu/~sbowman/multinli/

- **Size:** 433k sentence pairs
- **SoTA model:** T5-11B
- **Metric:** Accuracy

Transfer Learning with a Unified Text-to-Text Transformer

Trend on MNLI corpus

9 GLUE tasks, 8 Super-GLUE Tasks, 6 MRQA tasks

Figure 1: CA-MTL\textsubscript{BASE} architecture first uses our uncertainty-based sampling algorithm to choose task data for batching. Then, the input tokens go through a frozen embedding layer, followed by a Conditional Alignment layer. The rest contains frozen BERT-based Transformer layers and trainable adapters.
Grammatical Error Correction (GEC)

• Detecting and correcting grammatical errors from the sentence
  • Spelling errors
  • Wrong words
  • Bad grammars

Common Mistakes in English

- Lizzie approached me, say, "Hello!"
- Lizzie approached me and said, "Hello!"

- If someday we meet, would we again start?
- If someday we meet, would we start again?

- He neither has talent nor the desire to learn.
- He has neither talent nor the desire to learn.
Grammatical Error Correction (GEC)

• Neural machine translation approaches
  • Transformer and copy mechanism
  • 0.5880 $F_{0.5}$ score

Figure 1: Copy-Augmented Architecture.

Zhao et al., NAACL 2019
GEC Data Augmentation

- Generate wrong sentences from correct sentences
- Generated (wrong, correct) pairs are augmented to train the model
- The proposed methods can generate without annotated data
Model-based Data Generation

- Data generation model is trained with unlabeled data pair $\langle \tilde{y}, \tilde{y} \rangle$
- The model generates subtly wrong sentences
- Appropriate generated pairs are used to train the GEC model
Multi-domain Task-oriented Dialogue

• The system interacts with user to help the user achieve his/her goal
  • e.g. Restaurant reservation, hotel reservation...

• Specific domains & specific goals
  • ⇔ Open domain (chit-chat)

• Multi-turns

• User goals are not limited to just one domain

To book a hotel  ⇔  To book a restaurant near the hotel  ⇔  To book a taxi from the hotel to the restaurant
DST (Dialogue State Tracking)

• The most important subtask of task-oriented dialogue system

• Dialogue state generally consists of slot-value pairs
  • Slot means general class like food, area / value means specific value of slot

• In multi-domain, dialogue state consists of domain-slot-value triplets
  • ex) (restaurant-pricerange-cheap), (attraction-type-museum)...

• Infer dialogue state each turn of dialogues between user and system
  • There are ground truth dialogue states every turn

• Inferred state is used to generate next system action and response

• TRADE (TRAnsferable Dialogue statE generator)
  • 2019 ACL @ Hong Kong Univ. & Salesforce
    [https://www.aclweb.org/anthology/P19-1078]

• DS-DST (Dual Strategy for DST)
  • State-Of-The-Art in DST
  • 2019 Arxiv @ Illinois Univ. & Salesforce [https://arxiv.org/abs/1910.03544]
Dataset: MultiWOZ

• Standard benchmark dataset for end-to-end task-oriented dialogue systems
  • Multi-domain dataset for task-oriented dialog system
  • Virtual conversations between a tourist and a guide in Cambridge
  • 7 domains: Hotel, restaurant, attraction, train, taxi, police, hospital
  • Number of dialog: 10,000 / Avg. number of turns per dialog: 13

• Evaluation in MultiWOZ
  • Inform rate
    • How many entries provided by the system meet goals?
  • Success rate
    • How many user requests are provided by the system?
  • BLEU
    • Similarity between generated response and true response in the dataset
User: I am planning a trip to town and want to sight see a bit.
System: There are 79 attractions in the city. Do you have a specific type of attraction?

User: I’d like a architecture.
System: How about All Saints Church? It has fantastic architecture and free entrance.

User: Could I get the address? I would also like an expensive place to eat around it.
System: The address is Jesus Lane. What type of food would you like to eat?

User: I'm in the mood for Turkish food.
System: Anatolia is an expensive Turkish restaurant in the city centre.

User: Can I have the phone number?
System: The phone number is 01223362372.

User: Thank you!
System: You're welcome, feel free to contact us about anything else you need.
traditional pipeline architecture

User

“I want to find a Chinese restaurant.”

Natural Language Understanding

Inform (cuisine=“Chinese”) → Dialog State Tracking

Dialog Manager

Request (location)

Natural Language Generation

Dialog Policy

Knowledge Base

Query
Existing model: SOLOIST

- An auto-regressive model for training (Language modeling)
- Task 1: Predict dialogue state (slot-value)
- Task 2: Predict system response
- Task 3 (for Auxiliary loss)
  - Replace dialogue state or system response in input sequence into **negative samples** randomly
  - Then, predict input sequence is negative sample or not (binary classification)
- Jointly train by sum of 3 losses
Domain state tracking

- For tracking the flow of conversation from domain perspective
- Contains binary values indicating whether domains are activated or not in the current conversation
- Changes over turns

**User:** I would like to get some information about a restaurant and a hotel.

**System:** Okay, let's start with a hotel, any preference of type, area, or price range?

**User:** The hotel that I am looking for is called Gonville.

**System:** Do you want book the hotel?

...  

**User:** I want an Italian restaurant near the hotel.

**System:** How about Prezzo? It places centre city.

...
Architecture w/ domain state tracking

- NLU module is shared for DST, POL, and NLG
- Darker blocks mean previous turn
- DB result contains the number of matched entries for each domain

Cross entropy loss

Binary cross entropy loss
End-to-end ASR

Frontend (Preprocessing) STFT, MEL

Audio Inputs

Transformer Encoder

Speech Representation

Transformer Decoder (Attention)

CTC

Beam Search

Text Outputs

Short-Time Fourier Transform (STFT)

*Connectionist Temporal Classification (CTC)*
Conformer Encoder

Conformer (Encoder)

Decoder - LSTM


*convolution-augmented transformer for speech recognition, Conformer

*Specaughment: A simple data augmentation method for automatic speech recognition (to log-mel spectrogram): time warping, frequency & time masking
Conformer + Wav2Vec 2.0 (Encoder)
Decoder - LSTM

**Pre-training (wav2vec 2.0)**

1. Input Features
2. SpecAugment
3. Convolutional Subsampling
4. Linear
5. Conformer Blocks $\times N$
6. Masking
7. Encoded Features
8. Quantization
9. Feature Encoder
10. Context Network
11. Target Context Vectors
12. Contrastive Loss

**Standard Training**

1. Input Features
2. SpecAugment
3. Convolutional Subsampling
4. Linear
5. Dropout
6. Conformer Blocks $\times N$
7. Projection Block
8. Projected Context Vectors

*Pre-training: The contrastive loss between the context vectors from masked features and the quantization unit is optimized.

CTC-attention decoder

Joint CTC-Attention Decoding

LAS (Listen, Attend and Spell)

*accepts filter bank spectra as inputs

*Attention-based recurrent network decoder (character decoding)
*LAS does not make independence assumptions in the label sequence unlike CTC

Tacotron2: Seq2seq with attention RNN + modified WaveNet

*wavenet: invert the mel spectrogram feature representation into time-domain
waveform samples (10 component mixture of logistic distributions (MoL) to generate 16-bit samples at 24 kHz)

*location-sensitive attention: mitigating potential failure modes where some subsequences are repeated or ignored by the decoder
*Auto-regressive decoder to generate mel spectrogram frame by frame; Previous frame input to pre-net & post-net (for residual)
*stop token to stop generation
MelGan

Change WaveNet vocoder to:
- MelGan for faster inference

Figure 1: MelGAN model architecture. Each upsampling layer is a transposed convolution with kernel-size being twice of the stride (which is same as the upsampling ratio for the layer). 256x upsampling is done in 4 stages of 8x, 8x, 2x and 2x upsampling. Each residual dilated convolution stack has three layers with dilation 1, 3 and 9 with kernel-size 3, having a total receptive field of 27 timesteps. We use leaky-relu for activation. Each discriminator block has 4 strided convolution with stride 4. Further details can be found in the Appendix 6.
Dialog System for Counseling (TOD → Open-ended)

- Paraphrasing
- Reflect Feeling

- Correct Information
- Logical Answering

- Questioning
- Show Interest

- Listening
- Memorizing

COUNSELING

Empathy

Motivating

Sensibleness

Attending
CBT Counseling Scenario

Initiation
- Greeting
- Explanation of Counseling Method
- Extraction information of User

Situation
- 5W1H details
- Automatic thoughts
- Ground Belief
- Emotion
- Supporting Evidence
- No-Supporting Evidence

Refutation
- Alternative thoughts
- Emotion
- Motivating

*Automatic vs alternative
*For next visit
Variational hierarchical recurrent

VHRED: RNN Dialog Model

**Meena : Transformer Dialog Model**

User : How are things?

Meena : They're good. How about you?

User : I'm trying to decide what to watch

Meena : What are some of your favorite shows?

User : I like the Good Place

Meena : I haven't watched that. My favorite shows are Supernatural, and Star trek : TNG.

User : What is TNG?

**Evolved Transformer Encoder Block**

**Evolved Transformer Decoder Block**

Meena : The Next Generation

---

Frederick Parker-Rhodes (21 March 1914 - 21 November 1987) was an English linguist, plant pathologist, computer scientist, mathematician, mystic, and mycologist.

Data2Text NLG requires **accuracy (semantic fidelity)** as well as fluency.
Switch-GPT2 (Chen et al., 2020)

\[ L = L_c + \lambda \sum_{w_j \in m \atop m \in \{V_i\}} (1 - P^j_{\text{copy}}) \]

*Lc original loss w, Vᵢ slot-value, wⱼ target word in slot-value for copy*
• KGPT (Chen et al., 2020)

The encoder is pre-trained transformer model (specially trained for data2text task)

*Pretrained w/ (WikiData, WikiDump)

\[ P(y_i = w) = p_{gen}P_{voc}(w) + (1 - p_{gen}) \sum_{j: x_j = w} \alpha_j \]

*Fine tuned w/ any table2text
Both pre-trained D2T Architecture


table data

<table>
<thead>
<tr>
<th>Attention weights</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Pre-trained Encoder

RoBERTa

Table data

Pre-trained Language Model

GPT-2

Duplicate Copy penalty

\[
\text{loss}_i = -\log P(w_i^*) + \lambda \sum_i \min(d_i^t, c_i^t)
\]

\[
c^t = \sum_{t'=0}^{t-1} d^t'
\]

ToTTo dataset (Parikh et al., 2020): Wikipedia tables paired with natural language descriptions 120K